Separating OR, SUM, and XOR Circuits[☆]

Magnus Find[a], Mika Göös[b,c], Matti Järvisalo[c], Petteri Kaski[d],
Mikko Koivisto[c], Janne H. Korhonen[c]

[a] *Department of Mathematics and Computer Science, University of Southern Denmark, Denmark*
[b] *Department of Computer Science, University of Toronto, Canada*
[c] *HIIT & Department of Computer Science, University of Helsinki, Finland*
[d] *HIIT & Department of Information and Computer Science, Aalto University, Finland*

Abstract

Given a boolean n by n matrix A we consider arithmetic circuits for computing the transformation $x \mapsto Ax$ over different semirings. Namely, we study three circuit models: monotone OR-circuits, monotone SUM-circuits (addition of non-negative integers), and non-monotone XOR-circuits (addition modulo 2). Our focus is on *separating* these models in terms of their circuit complexities. We give three results towards this goal:

(1) We prove a direct sum type theorem on the monotone complexity of tensor product matrices. As a corollary, we obtain matrices that admit OR-circuits of size $O(n)$, but require SUM-circuits of size $\Omega(n^{3/2}/\log^2 n)$.
(2) We construct so-called *k-uniform* matrices that admit XOR-circuits of size $O(n)$, but require OR-circuits of size $\Omega(n^2/\log^2 n)$.
(3) We consider the task of *rewriting* a given OR-circuit as a XOR-circuit and prove that any subquadratic-time algorithm for this task violates the strong exponential time hypothesis.

Keywords:
arithmetic circuits, boolean arithmetic, idempotent arithmetic, monotone separations, rewriting

[☆] This work is an extended version of two preliminary conference abstracts [8, 14].

1. Introduction

A basic question in arithmetic complexity is to determine the minimum size of an arithmetic circuit that evaluates a linear map $x \mapsto Ax$. In this work we approach this question from the perspective of relative complexity by varying the circuit model while keeping the matrix A fixed, with the goal of separating different circuit models. That is, our goal is to show the existence of A that admit small circuits in one model but have only large circuits in a different model.

We will focus on boolean arithmetic and the following three circuit models. Our circuits consist of either

1. only $\vee$-gates (i.e., boolean sums; rectifier circuits),
2. only $+$-gates (i.e., integer addition; cancellation-free circuits), or
3. only $\oplus$-gates (i.e., integer addition mod 2).

These three types of circuits have been studied extensively in their own right (see Section 2), but fairly little is known about their relative powers.

Each model admits a natural description both from an algebraic and a combinatorial perspective.

Algebraic perspective. In the three models under consideration, each circuit with inputs $x_1, \ldots, x_n$ and outputs $y_1, \ldots, y_m$ computes a vector of *linear forms*

$$y_i = \sum_{j=1}^{n} a_{ij} x_j, \qquad i = 1, \ldots, m.$$

That is, $y = Ax$, where $A = (a_{ij})$ is an m by n boolean matrix with $a_{ij} \in \{0, 1\}$ and the arithmetic is either

1. in the boolean semiring $(\{0, 1\}, \vee, \wedge)$,
2. in the semiring of non-negative integers $(\mathbb{N}, +, \cdot)$, or
3. in GF(2).

As an example, Fig. 1 displays two circuits for computing $y = Ax$ for the same A using two different operators; the circuit on the right requires one more gate.

Combinatorial perspective. A circuit computing $y = Ax$ for a boolean matrix A can also be viewed combinatorially: every gate g is associated with a subset of the formal variables $\{x_1, \ldots, x_n\}$; this set is called the *support* of g and it is denoted $\mathrm{supp}(g)$. The input gates correspond to the singletons $\{x_j\}$, $j = 1, \ldots, n$, and every non-input gate computes either

1. the set union ($\vee$),
2. the disjoint set union ($+$), or
3. the symmetric difference ($\oplus$) of its children.

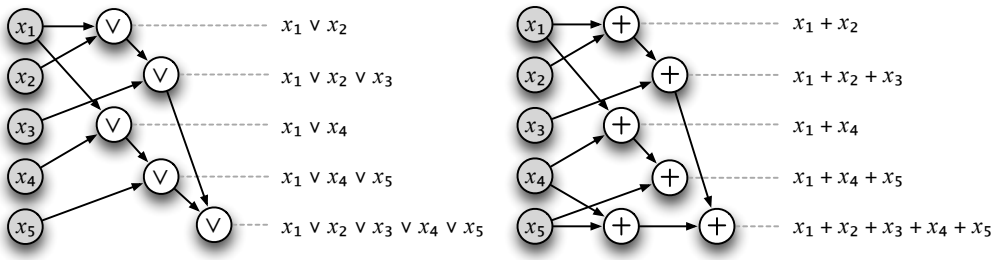

Figure 1: An $\vee$-circuit (left) and a $+$-circuit (right).

This way an output gate y_i will have $\mathrm{supp}(y_i) = \{x_j : a_{ij} = 1\}$.

Note the special structure of a $+$-circuit: there is at most one directed path from any input x_j to any output y_i. In fact, from this perspective, every $+$-circuit for A is easy to interpret both as an $\vee$-circuit for A, and as a $\oplus$-circuit for A (equivalently, there are onto homomorphisms from $(\mathbb{N}, +, \cdot)$ to $(\{0,1\}, \vee, \wedge)$ and $\mathrm{GF}(2)$). In this sense, both $\vee$- and $\oplus$-circuits are at least as efficient as $+$-circuits.

Relative complexity. More generally we fix a boolean matrix A and ask how the circuit complexity of computing $y = Ax$ depends on the underlying arithmetic.

To make this quantitative, denote by $C_\vee(A)$, $C_+(A)$, and $C_\oplus(A)$ the minimum number of wires in an unbounded fan-in circuit for computing $y = Ax$ in the respective models. For simplicity, we restrict our attention to the case of square matrices so that $m = n$.

For $\mathsf{X}, \mathsf{Y} \in \{\vee, +, \oplus\}$, we are interested in the complexity ratios

$$Gap_{\mathsf{X}/\mathsf{Y}}(n) \;:=\; \max_{A \in \{0,1\}^{n \times n}} C_{\mathsf{X}}(A)/C_{\mathsf{Y}}(A).$$

For example, we have that $Gap_{\vee/+}(n) = Gap_{\oplus/+}(n) = 1$ and that $Gap_{+/\oplus}(n) \geq Gap_{\vee/\oplus}(n)$ for all n, by the above fact that each $+$-circuit can be interpreted as an $\vee$-circuit and as a $\oplus$-circuit.

We review the motivation for studying separation bounds in Section 2. Next, we state our results, which are summarised in Figure 2.

1.1. Our results

We begin by studying the monotone complexity of tensor product matrices of the form

$$A = B_1 \otimes B_2,$$

where $\otimes$ denotes the usual Kronecker product of matrices. In Section 3, we prove a direct sum type theorem on their monotone complexity. As a corollary, we obtain matrices that are easy for $\vee$-circuits, $C_\vee(A) = O(n)$, but hard for $+$-circuits, $C_+(A) = \Omega(n^{3/2}/\log^2 n)$. This implies our first separation:

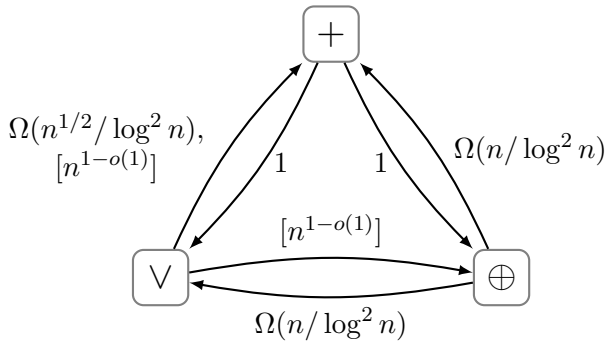

Figure 2: Separation bounds. An arrow from Y to X is labelled with $Gap_{\mathsf{X}/\mathsf{Y}}(n)$; bounds for (X,Y)-Rewrite are given inside square brackets.

Theorem 1. $Gap_{+/\vee}(n) = \Omega(n^{1/2}/\log^2 n)$.

We are not aware of any prior lower bound techniques that work against $+$-circuits, but not against $\vee$-circuits. Hence, as far as we know, Theorem 1 is a first step in this direction.

Next, we separate $\vee$- and $+$-circuits from $\oplus$-circuits by considering matrices that look *locally random* in the following sense:

Definition (k-uniformity)**.** A random matrix $\boldsymbol{A}$ is called *k-uniform* if the entries in every $k \times k$ submatrix have a marginal distribution that is uniform on $\{0,1\}^{k\times k}$.

Equivalently, a matrix is k-uniform if each of its entries is 0 or 1 with equal probability and the entries in every $k \times k$ submatrix are mutually independent.

In Section 4 we construct $n^{\Omega(1)}$-uniform matrices that are easy for $\oplus$-circuits:

Theorem 2. *There are $n^{\Omega(1)}$-uniform matrices $\boldsymbol{A}$ having $C_{\oplus}(\boldsymbol{A}) = O(n)$.*

These k-uniform matrices turn out to be difficult to compute using monotone circuits. Indeed, as a corollary, we will obtain our second separation:

Corollary 3. $Gap_{\vee/\oplus}(n), Gap_{+/\oplus}(n) = \Omega(n/\log^2 n)$.

Separations between $\vee$- and $\oplus$-circuits have also been considered by Sergeev et al. [11, 12] who proved the slightly weaker bound $Gap_{\vee/\oplus}(n) = \Omega(n/(\log^6 n \log\log n))$. Furthermore, Jukna [17] has informed us that the bound in Corollary 3 can actually be proved more directly using existing methods [15, 28]. Nevertheless, we hope our alternative approach via k-uniform matrices might be of independent interest—for example, in closing the gap between the current lower bound $Gap_{\vee/\oplus}(n) = \Omega(n/\log^2 n)$ and the best known upper bound $Gap_{\vee/\oplus}(n) = O(n/\log n)$; see Section 2.

As is true in the case of $Gap_{\vee/\oplus}$ we conjecture more generally that all the non-trivial complexity gaps between the three models are of order $n^{1-o(1)}$. While we are unable to enlarge the gap in Theorem 1, or prove any super-constant lower bounds on $\rho_{\oplus/\vee}$, our final result provides some evidence towards these conjectures.

In Section 5, we show that if certain $\vee$-circuits that are derived from CNF formulas could be efficiently *rewritten* as equivalent +- or $\oplus$-circuits, this would imply unexpected consequences for exponential-time algorithms. More precisely, we study the following problem.

The (X,Y)-Rewrite **problem:** On input an X-circuit $\mathcal{C}$, output a Y-circuit that computes the same matrix as $\mathcal{C}$.

Both $(\vee,+)$-Rewrite and $(\vee,\oplus)$-Rewrite admit simple algorithms that output a circuit of size $O(|\mathcal{C}|^2)$ in time $O(|\mathcal{C}|^2)$. However, we show that any significant improvement on these algorithms would give a non-trivial $2^{(1-\epsilon)n}\operatorname{poly}(n,m)$ time algorithm for deciding whether an n-variable m-clause CNF formula is satisfiable—this violates *the strong exponential time hypothesis* [13]:

Theorem 4. *Neither* $(\vee,+)$-Rewrite *nor* $(\vee,\oplus)$-Rewrite *can be solved in time* $O(|\mathcal{C}|^{2-\epsilon})$ *for any constant* $\epsilon>0$*, unless the strong exponential time hypothesis fails.*

Theorem 4 provides evidence, e.g., for the conjecture $\rho_{\oplus/\vee}=n^{1-o(1)}$ in the following sense. If there is a family of matrices A witnessing $C_\oplus(A)/C_\vee(A)=n^{1-o(1)}$, then clearly no $O(|\mathcal{C}|^{2-\epsilon})$-time algorithm exists for $(\vee,\oplus)$-Rewrite: if we are given a minimum-size $\vee$-circuit for A as input, there is no time to write down a legal output.

Our proof of Theorem 4 shows, in particular, that an $O(|\mathcal{C}|^{2-\epsilon})$-time algorithm for $(\vee,+)$-Rewrite would give an improved algorithm for counting the number of satisfying assignments to a given CNF formula (#CNF-SAT). Similarly, an $O(|\mathcal{C}|^{2-\epsilon})$-time algorithm for $(\vee,\oplus)$-Rewrite would give an improved algorithm for deciding whether the number of satisfying assignments is odd ($\oplus$CNF-SAT).

1.2. Notation

A *circuit* $\mathcal{C}$ is a directed acyclic graph where the vertices of in-degree (or *fan-in*) zero are called *input gates* and all other vertices are called *arithmetic gates*. One or more arithmetic gates are designated as *output gates*. The size $|\mathcal{C}|$ of the circuit is the number of edges (or *wires*) in the circuit.

We abbreviate $[n]:=\{1,\dots,n\}$; all our logarithms are to base 2 by default; and we write random variables in boldface.

2. Related work

Upper bounds. The trivial depth-1 circuit for a boolean matrix A uses $|A|$ wires, where we denote by $|A|$ the *weight* of A, i.e., the number of 1-entries in A. Even though $|A|$ might be of order $\Theta(n^2)$, Lupanov (as presented by Jukna [16, Lemma 1.2]) constructs depth-2 circuits (applicable in all the three models) of size $O(n^2/\log n)$ for any A. This implies the universal upper bound

$$Gap_{\mathsf{X}/\mathsf{Y}}(n)=O(n/\log n). \qquad \text{(Lupanov)}$$

Lower bounds. Standard counting arguments [16, §1.4] show that most $n \times n$ matrices have wire complexity $\Omega(n^2/\log n)$ in each of the three models. Combining this with Lupanov's upper bound we conclude that a random matrix does little to separate our models:

Fact 1. For a uniformly random $\boldsymbol{A}$, the ratio $C_{\mathsf{X}}(\boldsymbol{A})/C_{\mathsf{Y}}(\boldsymbol{A})$ is a constant w.h.p.

Unsurprisingly, it can also be shown that finding a minimum-size circuit for a given matrix is NP-hard in all the models. For $\vee$- and $+$-circuits this follows from the NP-completeness of the Ensemble Computation problem as defined by Garey and Johnson [10, PO9]. For $\oplus$-circuits this was proved by Boyar et al. [5].

$\vee$-circuits. The study of $\vee$-circuits (sometimes called *rectifier circuits*) has been centered around finding *explicit* matrices that are hard for $\vee$-circuits. Here, dense *rectangle-free* matrices and their generalisations, *(s,t)-free* matrices, are a major source of lower bounds.

Definition. A matrix A is called *(s,t)-free* if it does not contain an $(s+1) \times (t+1)$ all-1 submatrix. Moreover, A is simply called *k-free* if it is (k,k)-free.

Nechiporuk [24] and independently Lamagna and Savage [21] constructed the first examples of dense 1-free matrices A achieving $C_{\vee}(A) = \Omega(n^{3/2})$. Subsequently, Mehlhorn [22] and Pippenger [26] established the following theorem that gives a general template for this type of lower bound; we use it extensively later.

Theorem 5 (Mehlhorn–Pippenger)**.** *If A is (s,t)-free, then $C_{\vee}(A) \geq |A|/(st)$.*

Currently, the best lower bound for an explicit A is obtained by applying Theorem 5 to a matrix construction of Kollár et al. [19]; the lower bound is $C_{\vee}(A) \geq n^{2-o(1)}$ (see also Gashkov and Sergeev [11, §3.2]).

$\oplus$-circuits. It is a long-standing open problem to exhibit explicit matrices requiring super-linear size $\oplus$-circuits. No such lower bounds are known even for log-depth circuits, and the only successes are in the case of bounded depth [2, 9], [16, §13.5]. This, together with Fact 1, makes it particularly difficult to prove lower bounds on $Gap_{\oplus/\vee}$.

$+$-circuits. Additive circuits have been studied extensively in the context of the *addition chain* problem (see Knuth [18, §4.6.3] for a survey) and its generalisations [27].

In cryptography, as observed by Boyar et al. [5], many heuristics that have been proposed for finding small $\oplus$-circuits produce, in fact, $+$-circuits that do not exploit the cancellation of variables that is available in GF(2). Thus, the measure $Gap_{+/\oplus}$ gives a lower bound on the approximation ratio achieved by any such minimisation heuristic.

Algebraic complexity. A particular motivation for studying the separation between $\vee$- and $+$-circuits is to understand the complexity of zeta transforms on partial orders [3]. Indeed, the characteristic matrix of every partial order $\leq$ has an $\vee$-circuit proportional to the number of covering pairs in $\leq$, but the existence of small $+$-circuits (and hence fast zeta transforms) is not currently understood satisfactorily.

Strong exponential time hypothesis. Theorem 4 is similar to other recent lower bound results for polynomial-time solvable problems based on the strong exponential time hypothesis [25]. See also [7].

3. $+/\vee$-Separation

In this section we give a direct sum type theorem for the monotone complexity of tensor product matrices. Using this, we obtain a separation of the form

$$\begin{aligned} C_\vee(B \otimes A) &= O(N), \\ C_+(B \otimes A) &= \Omega(N^{3/2}/\log^2 N), \end{aligned} \tag{1}$$

where $\otimes$ denotes the usual Kronecker product of matrices and $N = n^2$ denotes the number of input and output variables. This will prove Theorem 1.

3.1. Tensor products

As a first example, let A be a fixed boolean $n \times n$ matrix and consider the matrix product

$$X \mapsto AX\,, \tag{2}$$

where we think of X as a matrix of $N = n \times n$ input variables. If we arrange these variables into a column vector x by stacking the columns of X on top of one another, then (2) becomes

$$x \mapsto (I \otimes A)x, \tag{3}$$

where I is the $n \times n$ identity matrix. That is, $I \otimes A$ is the block matrix having n copies of A on the diagonal.

The transformation (3) famously admits non-trivial $\oplus$-circuits due to the fact that fast matrix multiplication algorithms can be expressed as small bilinear circuits over GF(2). However, it is easy to see that in the case of our monotone models, no non-trivial speed-up is possible: any $\vee$-circuit for (3) must compute A independently n times:

$$C_\vee(I \otimes A) = n \cdot C_\vee(A). \tag{4}$$

This follows from the observation that two subcircuits corresponding to two different columns of X cannot share gates due to monotonicity.

Our approach. We will generalise the above setting slightly and use tensor products of the form $B \otimes A$ to separate $\vee$- and $+$-circuits. Analogously to (2), one can check that the matrix $B \otimes A$ corresponds to computing the mapping

$$X \mapsto AXB^\top. \tag{5}$$

We aim to show that for suitable choices of A and B computing $B \otimes A$ is easy for $\vee$-circuits but hard for $+$-circuits. We will choose A to have large complexity (e.g.,

choose A at random), and think of B as dictating how many independent copies of A a circuit must compute.

More precisely, define $\mathrm{rk}_\vee(B)$ and $\mathrm{rk}_+(B)$ as the minimum r such that B can be written as $B = PQ^\top$ over the boolean semiring or over the semiring of non-negative integers, respectively, where P and Q are $n \times r$ matrices. Equivalently, $\mathrm{rk}_\vee(B)$ (resp., $\mathrm{rk}_+(B)$) is the minimum number of rectangles (resp., non-overlapping rectangles) that are required to cover all 1-entries of B.

These cover numbers appear often in the study of communication complexity [20]. In this context, the matrix $B = \bar{I}$—the boolean complement of the identity I—is the usual example demonstrating a large gap between the two concepts [20, Example 2.5]:

$$\begin{aligned} \mathrm{rk}_\vee(\bar{I}) &= \Theta(\log n), \\ \mathrm{rk}_+(\bar{I}) &= n. \end{aligned}$$

We will use this gap to show that, up to polylogarithmic factors,

$$\begin{aligned} C_\vee(\bar{I} \otimes A) &\approx \mathrm{rk}_\vee(\bar{I}) \cdot n^2, \\ C_+(\bar{I} \otimes A) &\approx \mathrm{rk}_+(\bar{I}) \cdot n^2. \end{aligned}$$

In terms of the number of input variables $N = n^2$, we will obtain (1).

3.2. Upper bound for $\vee$-circuits

Suppose $B = PQ^\top$ where P and Q are $n \times \mathrm{rk}_\vee(B)$ matrices. We can compute (5) as

$$(A(XQ))P^\top,$$

which requires 3 matrix multiplications, each involving $\mathrm{rk}_\vee(B)$ as one of the dimensions (the other dimensions being at most n).

If these 3 multiplications are naively implemented with an $\vee$-circuit of depth 3, each layer will contain at most $\mathrm{rk}_\vee(B)n^2$ wires so that $C_\vee(B \otimes A) \le 3\,\mathrm{rk}_\vee(B)n^2$. However, one can still use Lupanov's techniques to save an additional logarithmic factor: if $\mathrm{rk}_\vee(B) = O(\log n)$, Corollary 1.35 in Jukna [16] can be applied to show that each of the three multiplications above can be computed using $O(n^2)$ wires. Thus, for $B = \bar{I}$ we get

Lemma 6. *$C_\vee(\bar{I} \otimes A) = O(n^2)$ for all A.* □

3.3. Lower bound for $+$-circuits

Intuitively, since low-rank decompositions are not available for $\bar{I}$ in the semiring of non-negative integers, a $+$-circuit for $\bar{I} \otimes A$ should be forced to compute $\mathrm{rk}_+(\bar{I}) = n$ independent copies of A. More generally, we ask

Direct sum question. Do we have $C_+(B \otimes A) \ge \mathrm{rk}_+(B) \cdot C_+(A)$ for all A, B?

Alas, we can answer this affirmatively only in some special cases. For example, the trivial case $B = I$ was discussed above (4), and it is not hard to generalise the argument to show that the lower bound holds in case B admits a fooling set of size $\mathrm{rk}_+(B)$. (When B is viewed as an incidence matrix of a bipartite graph, a *fooling set* is a matching no two of whose edges induce a 4-cycle. See [20, §1.3].) However, since this will not be the case when $B = \bar{I}$, we will settle for the following version, which suffices for the separation result.

Theorem 7. *For all (s,t)-free A,*

$$C_+(B \otimes A) \geq \mathrm{rk}_+(B) \cdot \frac{|A|}{st}. \tag{6}$$

Note that if we set $B = I$ in Theorem 7 we recover essentially Theorem 5.

For the purposes of the proof we switch to the combinatorial perspective: For A and B we introduce two sets of n formal variables X_A and X_B. Moreover, we let $A_1, \ldots, A_n \subseteq X_A$ and $B_1, \ldots, B_n \subseteq X_B$ denote the associated outputs. That is, each output A_i is defined by one row of A, and each output B_j is defined by one row of B. With this terminology, the input variables for $B \otimes A$ are the pairs in $X_A \times X_B$; we think of X_A as indexing the rows and X_B as indexing columns of the variable matrix $X_A \times X_B$. Finally, $B \otimes A$ corresponds to computing the n^2 outputs

$$A_i \times B_j, \quad \text{for } i, j \in [n].$$

In the following proof we use the (s,t)-freeness of A to "zoom in" on that layer of the circuit which reveals the large wire complexity (similarly to Mehlhorn [22]). We advise the reader to first consider the case $s = t = 1$, as this already contains the main idea of the proof.

Theorem 7. Let $\mathcal{C}$ be a $+$-circuit computing $B \otimes A$. As a first step, we simplify $\mathcal{C}$ by allowing input gates to have larger-than-singleton supports. Namely, let F consist of those gates of $\mathcal{C}$ whose supports are contained in a *t-wide row cylinder* of the form $Y \times X_B$ where $Y \subseteq X_A$ and $|Y| \leq t$. We simply declare that all computations done by gates in F come for free: we promote a gate in F to an input gate and delete all its incoming wires. We continue to denote the modified circuit by $\mathcal{C}$—clearly, these modifications only decrease its wire complexity.

Call a wire that is connected to an input gate an *input wire* and denote the set of input wires by W. The wire complexity lower bound (6) will follow already from counting the number $|W|$ of input wires.

For $i \in [n]$ denote by $\mathcal{C}_i$ the subcircuit of $\mathcal{C}$ computing the n outputs $A_i \times B_j$, $j \in [n]$, and denote by $W(i)$ the input wires of $\mathcal{C}_i$; we claim that

$$|W(i)| \geq \mathrm{rk}_+(B) \cdot \frac{|A_i|}{t}. \tag{7}$$

Before we prove (7), we note how it implies the theorem. Each input wire $w \in W$ is feeding into a non-input gate having their support not contained in a t-wide row

cylinder. Due to (s,t)-freeness of A this means that w can appear only in at most s different $\mathcal{C}_i$. Thus, the sum $\sum_i |W(i)|$ counts w at most s times and, more generally, we have

$$|W| = \left|\bigcup_{i=1}^{n} W(i)\right| \geq \sum_{i=1}^{n} \frac{|W(i)|}{s},$$

which implies (6) given (7).

Proof of (7). Fix $i \in [n]$. If A_i is empty the claim is trivial. Otherwise fix a variable $x \in A_i$ and consider the structure of $\mathcal{C}_i$ when restricted to the variables $\{x\} \times X_B$. Since this set of variables can be naturally identified with X_B by ignoring the first coordinate, we can view $\mathcal{C}_i$ as computing a copy of B on the variables $\{x\} \times X_B$.

Indeed, we define the *x-support* $\mathrm{supp}_x(w)$ of an input wire $w \in W(i)$ to be the set of $y \in X_B$ such that the variable (x,y) is contained in the support of w. (The support of w is simply the support of the adjacent input gate.) Moreover, we let

$$W_x(i) := \{w \in W(i) : \mathrm{supp}_x(w) \neq \varnothing\}.$$

Put otherwise, $W_x(i)$ consists of the input wires that are used by $\mathcal{C}_i$ in computing a copy of B on the variables $\{x\} \times X_B$. Associate to each $w \in W_x(i)$ a rectangle

$$R_x(w) := \text{co-supp}_x(w) \times \mathrm{supp}_x(w),$$

where $\text{co-supp}_x(w)$ is the set of $j \in [n]$ such that w appears in the subcircuit $\mathcal{C}_{ij}$ of $\mathcal{C}_i$ that computes the output $A_i \times B_j$. Now, the crucial observation is that the collection of rectangles $\{R_x(w) : w \in W_x(i)\}$ is a non-overlapping cover of B, because $\mathcal{C}_i$ computes a copy of B by taking disjoint unions of the supports $\{\mathrm{supp}_x(w) : w \in W_x(i)\}$. Therefore, we must have that

$$|W_x(i)| \geq \mathrm{rk}_+(B). \tag{8}$$

To finish the proof, we note that a single input wire $w \in W(i)$, being t-wide, can only be contained in the sets $W_x(i)$ for at most t different $x \in A_i$. Thus, the sum $\sum_x |W_x(i)|$ counts w at most t times and, more generally, we have

$$|W(i)| = \left|\bigcup_{x \in A_i} W_x(i)\right| \geq \sum_{x \in A_i} \frac{|W_x(i)|}{t},$$

which implies (7) given (8). □

As will be shortly discussed in Section 4.1, a random matrix $\boldsymbol{A} \in \{0,1\}^{n \times n}$ is $O(\log n)$-free and has weight $|\boldsymbol{A}| = \Theta(n^2)$ w.h.p. Using these facts we obtain the following corollary, which, together with Lemma 6, proves Theorem 1.

Corollary 8. *A random $\boldsymbol{A}$ satisfies $C_+(\bar{I} \otimes \boldsymbol{A}) = \Omega(n^3/\log^2 n)$ w.h.p.* □

4. $\vee/\oplus$-Separation

In this section we use the probabilistic method to construct k-uniform matrices $\boldsymbol{A}$ that, for large enough k, will witness the following complexity gap with high probability:

$$C_{\oplus}(\boldsymbol{A}) = O(n),$$
$$C_{\vee}(\boldsymbol{A}) = \Omega(n^2/\log^2 n).$$

In what follows, all matrix arithmetic will be over $\mathbb{F} = \mathrm{GF}(2)$.

4.1. Motivation for k-uniform matrices

Suppose first that $\boldsymbol{A} \in \mathbb{F}^{n\times n}$ is a random matrix where each entry is drawn uniformly and independently from $\mathbb{F}$. The probability that $\boldsymbol{A}$ fails to be $(k-1)$-free can be bounded from above by taking the union bound over all possible $k \times k$ submatrices:

$$\Pr[\,\boldsymbol{A} \text{ is not } (k-1)\text{-free}\,] \leq \binom{n}{k}^2 2^{-k^2}. \tag{9}$$

It is easy to check (and well-known in the context of random graphs [4, §11]) that for $k \geq 2\log n$ this quantity tends to 0 as $n \to \infty$.

Our key observation here is that the estimate (9) only uses the property that the entries in each $k \times k$ submatrix of $\boldsymbol{A}$ are mutually independent. Indeed, the above analysis holds even when $\boldsymbol{A}$ is only k-uniform for $k \geq 2\log n$. Thus, we have the following lemma.

Lemma 9. *If $\boldsymbol{A}$ is k-uniform for $k \geq 2\log n$, then w.h.p.,*

$$C_{\vee}(\boldsymbol{A}) = \Omega(n^2/\log^2 n).$$

Proof. Any 2-uniform matrix $\boldsymbol{A}$ has pairwise independent entries so that $|\boldsymbol{A}| = \Theta(n^2)$ w.h.p. by Chebyshev's inequality. On the other hand, the above discussion implies that $\boldsymbol{A}$ is $2\log n$-free w.h.p. Thus, the claim follows from Theorem 5. □

Corollary 3 is a consequence of Lemma 9 and Theorem 2. Thus, our remaining goal in this section is to prove Theorem 2.

4.2. Proof of Theorem 2

Let $m := O(\sqrt{n})$. To construct a k-uniform matrix $\boldsymbol{A}$ we start with an $m \times n$ matrix P that satisfies the following two properties:

(1) P has linear $\oplus$-complexity, $C_{\oplus}(P) = O(n)$.
(2) Each set of $k = n^{\Omega(1)}$ columns of P are linearly independent.

Miltersen [23] shows that such P can be obtained as submatrices of certain generating matrices of linear codes, e.g., those of Spielman [29].

Theorem 10 (Miltersen [23, Theorem 1.4])**.** *Let $D \subseteq \mathbb{F}^n$. There are $O(\log |D|) \times n$ matrices P with $C_\oplus(P) = O(n)$ such that the mapping $x \mapsto Px$ is injective on D.*

Indeed, let $D \subseteq \mathbb{F}^n$ be the set of vectors of Hamming weight at most k. Note that if P is injective on D, then it clearly has property (2). We also have that $|D| \leq (n+1)^k$ and so $\log |D| = O(k \log n)$. Thus, if we set $k := \sqrt{n}/\log n$, we can apply Theorem 10 to obtain our desired $m \times n$ matrix P.

We can now define

$$\boldsymbol{A} := P^\top \boldsymbol{R} P,$$

where $\boldsymbol{R} \in \mathbb{F}^{m \times m}$ is a matrix chosen uniformly at random; note that $C_\oplus(\boldsymbol{R}) \leq |\boldsymbol{R}| \leq m^2 = O(n)$. If we compute $\boldsymbol{A}$ in three stages in the obvious way, we obtain

$$C_\oplus(\boldsymbol{A}) \leq C_\oplus(P^\top) + C_\oplus(\boldsymbol{R}) + C_\oplus(P) = O(n),$$

where we used the fact that $C_\oplus(P^\top) = O(C_\oplus(P))$—roughly, this follows from simply reversing the direction of the wires in a $\oplus$-circuit computing P (see Jukna [16, p. 46]).

It remains to show that $\boldsymbol{A}$ is k-uniform. In fact, since our definition of $\boldsymbol{A}$ is a generalisation of how k-wise independent variables are typically constructed [1, §15.2], the proof of the following lemma is somewhat routine.

Lemma 11. *$\boldsymbol{A}$ is k-uniform.*

Proof. We need to show that each submatrix $\boldsymbol{A}_{I \times J}$, where $I, J \subseteq [n]$ and $|I| = |J| = k$, is uniformly distributed in $\mathbb{F}^{k \times k}$. Write

$$\boldsymbol{A}_{I \times J} = {P_I}^\top \boldsymbol{R} P_J,$$

where P_K is the submatrix of P consisting of the columns with indices in $K \subseteq [n]$.

Claim. $\boldsymbol{B} := \boldsymbol{R} P_J$ is uniformly distributed in $\mathbb{F}^{m \times k}$.

Proof of Claim. Let $\boldsymbol{B}_i = \boldsymbol{R}_i P_J$ denote the i-th row of $\boldsymbol{B}$. The rows $\boldsymbol{B}_i$, $i \in [m]$, are mutually independent variables, since the variables $\boldsymbol{R}_i$, $i \in [m]$, are. Therefore it suffices to show that $\boldsymbol{B}_i$ is uniformly distributed in $\mathbb{F}^{1 \times k}$ for each $i \in [m]$.

To this end, fix $i \in [m]$; we show that all the outcomes $\boldsymbol{B}_i = y$ where $y \in \mathbb{F}^{1 \times k}$ are equally likely. For any $y \in \mathbb{F}^{1 \times k}$ there is a vector $x \in \mathbb{F}^{1 \times m}$ with $xP_J = y$ since P_J has linearly independent columns. Hence $\boldsymbol{R}_i P_J = y$ iff $(\boldsymbol{R}_i - x)P_J = 0$. But $\boldsymbol{R}_i - x$ is distributed the same as $\boldsymbol{R}_i$ so that $\Pr[\boldsymbol{R}_i P_J = y] = \Pr[\boldsymbol{R}_i P_J = 0]$ is independent of the choice of y, as desired. ◇

Finally, the same analysis as above demonstrates that $\boldsymbol{A}_{I \times J} = {P_I}^\top \boldsymbol{B}$ is uniformly distributed in $\mathbb{F}^{k \times k}$ proving the lemma. □

Remark. Interestingly, Theorem 5 is unable to prove a better lower bound than $C_\vee(A) = \Omega(n^2/\log^2 n)$ for any matrix A. Is it true that for every $n^{\Omega(1)}$-uniform $\boldsymbol{A}$, we have that $C_\vee(\boldsymbol{A}) = \Theta(n^2/\log n)$ w.h.p.? A positive answer would give the tight bound $Gap_{\vee/\oplus}(n) = \Theta(n/\log n)$.

5. Rewriting

In this section we study what would happen if $(\vee,+)$-Rewrite or $(\vee,\oplus)$-Rewrite could be solved in subquadratic time. Namely, we show that this eventuality would contradict the strong exponential time hypothesis. This will prove Theorem 4. As discussed in Section 1.1, we interpret this as evidence for our conjectures $\rho_{+/\vee} = n^{1-o(1)}$ and $\rho_{\oplus/\vee} = n^{1-o(1)}$.

5.1. Preliminaries

For purposes of computations, we tacitly assume that $|\mathcal{C}| \geq n$ for any n-input circuit $\mathcal{C}$ considered in this section. This is to make each $\mathcal{C}$ admit a binary representation of length $\tilde{O}(|\mathcal{C}|)$ where the $\tilde{O}$ notation hides factors polylogarithmic in n. For concreteness, $\mathcal{C}$ might be represented as two lists: (i) the list of gates in $\mathcal{C}$, with output gates indicated, and (ii) the list of wires in $\mathcal{C}$; both lists are given in topological order, with the input wires of each gate forming a consecutive sublist of the list of wires. Whatever the encoding, we assume it is efficient enough so that the following property holds.

Proposition 12. *On input an* X*-circuit* $\mathcal{C}$ *and a vector* x*, the output* $\mathcal{C}(x)$ *can be computed in time* $\tilde{O}(|\mathcal{C}|)$ *(in the usual RAM model of computation).* □

The following proposition records a similar observation for circuit rewriting.

Proposition 13. *Both* $(\vee,+)$-Rewrite *and* $(\vee,\oplus)$-Rewrite *can be solved in time* $\tilde{O}(|\mathcal{C}|^2)$.

Proof. Suppose we are given an $\vee$-circuit $\mathcal{C}$ as input. The matrix A computed by $\mathcal{C}$ can be easily extracted from $\mathcal{C}$ in time $\tilde{O}(|\mathcal{C}|^2)$. We then simply output the trivial depth-1 $+$-circuit for A that has size at most $n^2 \leq |\mathcal{C}|^2$. □

5.2. Proof of Theorem 4

The main technical ingredient in our proof is Lemma 14 below, which states that if subquadratic-time rewriting algorithms exist, then certain simple covering problems can be solved faster than in a trivial manner.

In the following we consider set systems defined by $L_1,\ldots,L_n$ and $R_1,\ldots,R_n$ that are (not necessarily distinct) subsets of $[m]$. We say that (i,j) is a *covering pair* if $L_j \cup R_i = [m]$.

Lemma 14. *Suppose we are given sets* $L_1,\ldots,L_n,R_1,\ldots,R_n \subseteq [m]$ *as input.*

(a) If $(\vee,+)$-Rewrite *can be solved in time* $\tilde{O}(|\mathcal{C}|^{2-\epsilon})$ *for some constant* $\epsilon > 0$*, then the number of covering pairs can be computed in time* $\tilde{O}((nm)^{2-\epsilon})$.

(b) If $(\vee,\oplus)$-Rewrite *can be solved in time* $\tilde{O}(|\mathcal{C}|^{2-\epsilon})$ *for some constant* $\epsilon > 0$*, then the parity of the number of covering pairs can be computed in time* $\tilde{O}((nm)^{2-\epsilon})$.

Proof of (a). Let $A = (a_{ij})$ be an $n \times n$ matrix defined by $a_{ij} = 1$ iff (i, j) is a covering pair. We show how to compute $|A|$ without constructing A explicitly.

Suppose for a moment that we had a small $+$-circuit $\mathcal{C}$ for A. The value $|A|$ can be recovered from the circuit $\mathcal{C}$ in time $\tilde{O}(|\mathcal{C}|)$ via the following trick: evaluate $\mathcal{C}$ (over the integers) on the all-1 vector $\mathbb{1}$ to obtain $y = \mathcal{C}(\mathbb{1}) \in \mathbb{N}^n$; but now

$$|A| = \mathbb{1}^\top A \mathbb{1} = \mathbb{1}^\top \mathcal{C}(\mathbb{1}) = y_1 + \cdots + y_n. \tag{10}$$

Unfortunately, we do not know how to construct a small $+$-circuit for A. Instead, our key observation below will be that *the complement matrix* $\bar{A}$ admits an $\vee$-circuit $\mathcal{C}^\vee$ of size only $|\mathcal{C}^\vee| = O(nm)$. By assumption, we can then rewrite $\mathcal{C}^\vee$ as a $+$-circuit $\mathcal{C}^+$ in time $\tilde{O}(|\mathcal{C}^\vee|^{2-\epsilon}) = \tilde{O}((nm)^{2-\epsilon})$. In particular, the size of the new circuit must also be

$$|\mathcal{C}^+| = \tilde{O}\left((nm)^{2-\epsilon}\right).$$

Analogously to (10) we can then recover $|A|$ from $\mathcal{C}^+$ in time $\tilde{O}(|\mathcal{C}^+|)$:

$$|A| = n^2 - |\bar{A}| = n^2 - \mathbb{1}^\top \mathcal{C}^+(\mathbb{1}).$$

Indeed, it remains to describe how to construct $\mathcal{C}^\vee$ for $\bar{A}$ in time $\tilde{O}(nm)$.

Construction. Define a depth-2 circuit $\mathcal{C}^\vee$ follows: The 0-th layer of $\mathcal{C}^\vee$ hosts input gates l_j, $j \in [n]$; the 1-st layer contains intermediate gates g_k, $k \in [m]$; and the 2-nd layer contains output gates r_i, $i \in [n]$. Each input gate l_j is connected to gates g_k for $k \in [m] \setminus L_j$; similarly, each output gate r_i is connected to gates g_k for $k \in [m] \setminus R_i$. To see that $\mathcal{C}^\vee$ computes $\bar{A}$ note that there is a path from input l_i to output r_j iff there is a $k \in [m]$ such that $k \notin L_i \cup R_j$ iff (i, j) is not a covering pair. Note also that $|\mathcal{C}^\vee| \leq 2nm$ and that the construction takes time $\tilde{O}(nm)$. □

Proof of (b). The proof is the same as above, except we work over GF(2). □

Next, we reduce #CNF-SAT and ⊕CNF-SAT to the covering problems in Lemma 14. Here we are essentially applying a technique of Williams [30, Theorem 5].

Theorem 15. *We have the following reductions:*

(a) If $(\vee, +)$-Rewrite *can be solved in time* $\tilde{O}(|\mathcal{C}|^{2-\epsilon})$ *for some* $\epsilon > 0$*, then* #CNF-SAT *can be solved in time* $2^{(1-\epsilon/2)n} \operatorname{poly}(n, m)$.

(b) If $(\vee, \oplus)$-Rewrite *can be solved in time* $\tilde{O}(|\mathcal{C}|^{2-\epsilon})$ *for some* $\epsilon > 0$*, then* ⊕CNF-SAT *can be solved in time* $2^{(1-\epsilon/2)n} \operatorname{poly}(n, m)$.

Proof. Let $\varphi = \{C_1, \ldots C_m\}$ be an instance of CNF-SAT over variables $x_1, \ldots, x_n$. Without loss of generality (by inserting one variable as necessary), we may assume that n is even. Call the variables $x_1, \ldots, x_{n/2}$ *left* variables and the variables $x_{n/2+1}, \ldots, x_n$ *right* variables.

For each truth assignment $s \in \{0,1\}^{n/2}$ to the left variables, let $L_s \subseteq \varphi$ be the set of clauses satisfied by s. Similarly, for assignment $t \in \{0,1\}^{n/2}$ to the right variables, let $R_t \subseteq \varphi$ be the set of clauses satisfied by t. Clearly, the compound assignment (s,t) to all the variables satisfies φ if and only if $L_s \cup R_t = \varphi$. That is, the number of satisfying assignments is precisely the number of covering pairs of the set system $\{L_s, R_t\}$, $s,t \in \{0,1\}^{n/2}$. Thus, both claims follow from Lemma 14. □

We can now finish the proof of Theorem 4:

- For $(\vee,+)$-Rewrite the result follows immediately from Theorem 15.
- For $(\vee,\oplus)$-Rewrite we need to make the following additional argument. As discussed by Cygan et al. [7] the k-CNF Isolation Lemma of Calabro et al. [6] can be applied to show that any $2^{(1-\epsilon)n}\operatorname{poly}(n,m)$ time algorithm for $\oplus$CNF-SAT can be turned into an $2^{(1-\epsilon')n}\operatorname{poly}(n,m)$ time Monte Carlo algorithm for CNF-SAT where $\epsilon' > 0$. Recognising this, the result follows from Theorem 15.

Acknowledgements. We are grateful to Stasys Jukna for pointing out a more direct proof of Corollary 3 as referenced in the text. We also thank Igor Sergeev for providing many references, in particular, one simplifying our proof of Theorem 2. Furthermore, we thank Jukka Suomela for discussions.

This research is supported in part by Academy of Finland, grants 132380 and 252018 (M.G.), 252083 and 256287 (P.K.), and by Helsinki Doctoral Programme in Computer Science - Advanced Computing and Intelligent Systems (J.K.).

References

[1] N. Alon and J. H. Spencer. *The Probabilistic Method*. John Wiley & Sons, 2 edition, 2000.

[2] N. Alon, M. Karchmer, and A. Wigderson. Linear circuits over GF(2). *SIAM Journal on Computing*, 19(6):1064–1067, 1990. doi:10.1137/0219074.

[3] A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, J. Nederlof, and P. Parviainen. Fast zeta transforms for lattices with few irreducibles. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012)*, pages 1436–1444. SIAM, 2012.

[4] B. Bollobás. *Random Graphs*. Number 73 in Cambridge studies in advanced mathematics. Cambridge University Press, 2nd edition, 2001.

[5] J. Boyar, P. Matthews, and R. Peralta. Logic minimization techniques with applications to cryptology. *Journal of Cryptology*, 26:280–312, 2013. doi:10.1007/s00145-012-9124-7.

[6] C. Calabro, R. Impagliazzo, V. Kabanets, and R. Paturi. The complexity of unique k-SAT: An isolation lemma for k-CNFs. *Journal of Computer and System Sciences*, 74(3):386–393, 2008. doi:10.1016/j.jcss.2007.06.015.

[7] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlstrom. On problems as hard as CNF-SAT. In *Proceedings of the 27th Conference on Computational Complexity (CCC 2012)*, pages 74–84. IEEE, 2012. doi:10.1109/CCC.2012.36.

[8] M. G. Find, M. Göös, P. Kaski, and J. H. Korhonen. Separating OR, SUM, and XOR circuits. Submitted, 2013.

[9] A. Gál, K. A. Hansen, M. Koucký, P. Pudlák, and E. Viola. Tight bounds on computing error-correcting codes by bounded-depth circuits with arbitrary gates. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, pages 479–494. ACM, 2012. doi:10.1145/2213977.2214023.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[11] S. B. Gashkov and I. S. Sergeev. On the complexity of linear Boolean operators with thin matrices. *Journal of Applied and Industrial Mathematics*, 5:202–211, 2011. doi:10.1134/S1990478911020074.

[12] M. I. Grinchuk and I. S. Sergeev. Thin circulant matrixes and lower bounds on complexity of some Boolean operators. *Diskretnyĭ Analiz i Issledovanie Operatsiĭ*, 18:38–53, 2011.

[13] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

[14] M. Järvisalo, P. Kaski, M. Koivisto, and J. H. Korhonen. Finding efficient circuits for ensemble computation. In *Proceedings of the 15th International Conference on Theory and Applications of Satisfiability Testing (SAT 2012)*, pages 369–382. Springer, 2012. doi:10.1007/978-3-642-31612-8_28.

[15] S. Jukna. Disproving the single level conjecture. *SIAM Journal on Computing*, 36 (1):83–98, 2006. doi:10.1137/S0097539705447001.

[16] S. Jukna. *Boolean Function Complexity: Advances and Frontiers*, volume 27 of *Algorithms and Combinatorics*. Springer, 2012.

[17] S. Jukna. Comment on XOR versus OR circuits, April 2013. URL http://www.thi.informatik.uni-frankfurt.de/~jukna/boolean/comment9.html.

[18] D. E. Knuth. *The Art of Computer Programming*, volume 2. Addison–Wesley, 3rd edition, 1998.

[19] J. Kollár, L. Rónyai, and T. Szabó. Norm-graphs and bipartite Turán numbers. *Combinatorica*, 16(3):399–406, 1996. doi:10.1007/BF01261323.

[20] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[21] E. A. Lamagna and J. E. Savage. Computational complexity of some monotone functions. In *IEEE Conference Record of 15th Annual Symposium on Switching and Automata Theory*, pages 140–144, 1974. doi:10.1109/SWAT.1974.9.

[22] K. Mehlhorn. Some remarks on Boolean sums. *Acta Informatica*, 12:371–375, 1979. doi:10.1007/BF00268321.

[23] P. B. Miltersen. Error correcting codes, perfect hashing circuits, and deterministic dynamic dictionaries. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 1998)*, pages 556–563. SIAM, 1998.

[24] É. I. Nechiporuk. On a Boolean matrix. *Systems Theory Research*, 21:236–239, 1971.

[25] M. Pătraşcu and R. Williams. On the possibility of faster SAT algorithms. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2010)*, pages 1065–1075. SIAM, 2010.

[26] N. Pippenger. On another Boolean matrix. *Theoretical Computer Science*, 11(1): 49–56, 1980. doi:10.1016/0304-3975(80)90034-1.

[27] N. Pippenger. On the evaluation of powers and monomials. *SIAM Journal on Computing*, 9(2):230–250, 1980. doi:10.1137/0209022.

[28] P. Pudlák and V. Rödl. Pseudorandom sets and explicit constructions of Ramsey graphs. In *Complexity of computations and proofs*, volume 13 of *Quaderni Di Matematica*. 2004.

[29] D. A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1731, 1996. doi:10.1109/18.556668.

[30] R. Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2–3):357–365, 2005. doi:10.1016/j.tcs.2005.09.023.